REMARKS/ARGUMENTS

Claim 1 is pending in the present application. Claim 1 has been canceled. Claims 2-20 were previously canceled. New claim 21 is presented. Reconsideration of the claims is respectfully requested.

I.   35 U.S.C. § 101

The Examiner has rejected claim 1 under 35 U.S.C. § 101 as being directed towards non-statutory subject matter. This rejection is respectfully traversed. Claim 1 has been canceled. New claim 21 has been written to overcome the 35 U.S.C. § 101 rejection of previous claim 1. Support for the relevant portions of new claim 21 is found in the specification and drawings as follows: FIG. 1, FIG. 2, and paragraphs [0013] and [0021] through [0025].

II.   35 U.S.C. § 103, Obviousness

The Examiner has rejected claim 1 under 35 U.S.C. § 103 as being unpatentable over Narayanaswamy et al., U.S. Patent No. 7,069,553 B2 (hereinafter Narayanaswamy); in view of Chan et al., U.S. Patent No. 6,633,892 B1 (hereinafter Chan); further in view of Jackson, U.S. Patent No. 6,920,630 B2 (hereinafter Jackson). Office Action pp. 3-6. This rejection is respectfully traversed.

A. New Claim 21

Claim 1 has been cancelled and new claim 21 is presented. Support for new claim 21 is found in the specification and drawings as follows:

A computer implemented method for updating an original plurality of files within an original archive file without altering an order and an arrangement of the original plurality of files, comprising:	[0006], lines 1-2
using a computer having a processor and a memory connected to the processor, storing a program in the memory of the computer, the program adapted to cause the processor to perform steps comprising:	FIG. 1; FIG. 2, 106, 100, 200; [0013]
creating a temporary directory;	FIG. 3, 204; [0006], line 4; [0026], lines 4-5
copying the original archive file into the temporary directory;	FIG. 3, 206; [0026], lines 8-9
recording an order and an arrangement of the original archive file in a structure file in the temporary directory;	FIG. 3, 208; [0006], line 5; [0018]; [0026], lines 8-9
decompressing the plurality of original files within the archive file to the temporary directory;	FIG. 3, 210; [0006], line 5-6; [0014]; [0026], lines 10-11.
creating an updated plurality of files by accepting a	FIG. 3, 212, 214; [0006], line 6-7; [0026], lines 12-

user specification of a file of the original plurality of files, a field in the file, an old value, and a new value, searching the file and field for each occurrence of the old value, and replacing each occurrence of the old value with the new value;	15
compressing the updated plurality of files into a new archive file according to the order and the arrangement stored in the structure file, wherein the order and the arrangement is the same before and after the updated plurality of files are created.	[0012] and [0014] (definitions of archive and extract); definition of JAR file structure (archive file structure) in paragraph [0018], in lines 8-9 of paragraph [0006], and further in lines 21-22 of paragraph [0026]; FIG. 3, 218; [0006], lines 7-10; [0026], line 20-22.

B. Summary of Argument

The embodiment of claim 21 allows a user to decompress an original plurality of files from an archive file, make changes to the original plurality of files to create an updated plurality of files, and then compress the updated plurality of files into the archive file in the order and arrangement of the original archive file. Applicant submits that it is the function of claim 21 to change the values in the decompressed files and then compress the files back into the archive file without altering the archive file structure that is novel. Specifically, claim 21 describes this feature by reciting that the "order and the arrangement is the same before and after the updated plurality of files are created." This result could not be obtained without saving the archive file structure to the temporary file for use after the files in the archive file were updated. Then, once the files are updated, the files are compressed into a new archive file with the same structure as the original files had in the old archive file.

The prior art discloses the following:

- updating deployment descriptors in a JAR file for target servers (Narayanaswamy 2:38-41)

- extracting and modifiying one or more deployment descriptors (Narayanaswamy 6:36-39)

- opening an input JAR file and extracting files, and "storing of the byte representations of the files in the collection of JAR entity meta objects 42." (Chan, col. 7, lines 31-40, col. 7, lines 66 -col. 8, line 15.)

- specifying a temporary directory "by a code generator, eventually to be added to the input.jar file 46." (Chan 8:8)

None of the above references, individually or in combination, use a temporary directory so that a plurality of files can be decompressed, modified, and compressed into a new archive file with the same

order and arrangement as in the old archive file because of the order and arrangement saved in the temporary directory.

As will be explained further below, the prior art disclosures do not render claim 21 obvious. At the center of the dispute with the Examiner is whether the archive file structure remains the same before and after the files in the archive file structure are modified. The Examiner cited Narayanaswamy, col. 7, lines 58-61 and col. 8, lines 12-16 and stated "EN: the EAR file is repackaged with the modified deployment descriptors where the JAR file content must be archived according to the original JAR file construct since only the deployment descriptors are modified." Furthermore, the Examiner cites Narayanaswamy, col. 7, lines 58-61 and col. 8, lines 12-16. The Examiner cites Fig. 7, Fig. 8, col. 2, lines 38-41, col. 5, lines 23-35, col. 5, lines 57-col. 16, lines 38-63, and states "EN: the JAR file structure remains the same before and after the data is changed because only the data within the deployment descriptor is changed as a result of the update."

The Examiner appears to argue that modifying a deployment descriptor and placing it in an EAR file is the same as modifying files in an archieve file. The Examiner is either incorrectly interpreting the references or arguing that the result is inherent in the references. In either situation, the Examiner is wrong because JAR files typically include one or more J2EE modules making up a J2EE application... and "[e]ach J2EE module typically includes a corresponding [a] deployment descriptor that contains declarative data required during the deployment of the components in the module." (Narayanaswamy, col 5: 57-63). However, archiving involves compressing and decompressing files and in claim 21, then creating a new archive file with the same order and arrangement as the original archive file.

The references cited by the Examiner simply do not disclose the limitations of claim 21 when claim 21 is read a whole because it is the order and arrangement of the files that remains unchanged. The cited references are silent as to the order and arrangement remaining the same. All that is disclosed is a deployment descriptor being modified and then placed in a file. Moreover, files are not compressed into a new archive file.

C. "A computer implemented method for updating an original plurality of files within an original archive file without altering an order and an arrangement of the original plurality of files, comprising"

The previous claim 1, to which the Examiner responded, stated: "[a] method for updating application configuration information in a plurality of files within an archive file." The Examiner cited Narayanaswamy, col. 2, lines 38-41 and col. 5, lines 57-63 which state:

> Deployment files are updated with the determined configuration information and the program modules are repackaged with the update deployment files. (2:38-41)

JAR files typically include one or more J2EE modules making up a J2EE application. A J2EE module is a collection of one or more J2EE components of the same component type such as a web and EJB. Each J2EE module typically includes a corresponding deployment descriptor that contains declarative data required during the deployment of the components in the module. (5:57-63)

The preamble of new claim 21 recites that a result of the method is to update a plurality of files within an original archive file without altering an order and an arrangement of the original plurality of files. Narayanaswamy is silent as to updating a plurality of files without altering an order and an arrangement of the plurality of files.

D. "using a computer having a processor and a memory connected to the processor, storing a program in the memory of the computer, the program adapted to cause the processor to perform steps comprising: creating a temporary directory; copying the original archive file into the temporary directory; recording an order and an arrangement of the original archive file in a structure file in the temporary directory"

Previous claim 1, to which the Examiner responded, recited "recording an archive file structure of the archive file in a structure file in the temporary directory." In response, the Examiner cited Narayanaswamy FIG. 7; col. 6, lines 36-39, and col. 16, lines 38-44, and stated "EN: the JAR file structure must be recorded since EJB display tree containing the JAR file structure is displayed." Narayanaswamy states:

At 204, input EAR file 202 is processed. For example, the EAR file 202 is expanded into its individual components so that one or more deployment descriptors in the EAR file 202 may be extracted and modified. (6:36-39)

Once the deployment details have been specified for a server, the EAR file is packaged and displayed. The user may then modify the deployment descriptors for the J2EE modules. The deployment descriptor may be modified in a number of different ways. An XML editor may be used to edit the XML files selected, for example, from a panel showing EJB display tree illustrated in FIG. 7. (16:38-44)

New claim 21 recites "recording an order and an arrangement of the original archive file in a structure file in the temporary directory." Narayanaswamy is silent as to recording an order and arrangement of the original archive file in a structure file in the temporary directory.

The Examiner admitted that Narayanaswamy did not teach creating a temporary directory and copying an archive file into the temporary directory. The Examiner stated that Chan "teaches a deployment tool that creates a temporary directory and copies an archive file into the temporary directory to be processed." The Examiner cites Chan, Fig. 3, col. 7, lines 10-14 and 31-40. The Examiner states that it would have been obvious to one or ordinary skill in the art at the time of the invention to have

modified Narayanaswamy to create a temporary directory and copy the archive file into the temporary directory as taught by Chan because it provides a temporary working directory for working with the input Jar file." The Examiner cites Chan, col. 7, lines 31-40, col. 7, lines 66 -col. 8, line 15. The cited portions of Chan disclose opening an input JAR file and extracting files, and the "storing of the byte representations of the files in the collection of JAR entity meta objects 42." Storing byte representations of the files in the collection of JAR entity meta objects is not the same as "recording an order and an arrangement of the original archive file in a structure file in the temporary directory. At 8:8, Chan states "a temporary directory may be specified at Step 66 by a code generator, eventually to be added to the input.jar file 46." Chan's "temporary directory" is not the same as the temporary directory of claim 21 because it is permissive by virtue of the word "may" and Chan's temporary directory is destined for an input.jar file.

E. "decompressing the plurality of original files within the archive file to the temporary directory"

Former claim 1, to which the Examiner responded, recited "extracting the plurality of files within the archive file to the temporary directory." The Examiner stated that "i.e., one or more deployment descriptors in the EAR file may be extracted and modified." The Examiner further cites Fig. 7, Fig. 8, column 6, lines 36-48, and col. 8, lines 39-42. The cited portions of Narayanaswamy disclose extracting and modifying one or more deployment descriptors where a deployment descriptor is "an XML file provided for each module and application, and describes how the modules and applications are to be deployed." (6:36-48). New claim 21 recites "decompressing the plurality of original files within the archive file to the temporary directory." Narayanaswamy is silent as to "decompressing the plurality of original files within the archive file to the temporary directory."

F. "creating an updated plurality of files by accepting a user specification of a file of the original plurality of files, a field in the file, an old value, and a new value, searching the file and field for each occurrence of the old value, and replacing each occurrence of the old value with the new value"

Previous claim 1, to which the Examiner responded, recited "updating application configuration information in the plurality of files to create an updated plurality of files by performing a series of steps comprising: accepting a user specification of a file of the plurality of files, a field, an old value, and a new value; searching the field in the file for the old value; and replacing the old value with the new value." The Examiner cited Narayanaswamy, col. 8, lines 37-53, and col. 11, lines 7-41 stating that "'[a] string that may need to be replaced' is the old value and 'the replacement string' is the new value." The Examiner further stated: "[t]he checkForStart and CheckFor End methods of the Replacer interface are

called to locate the beginning and end of the string that may need to be replaced." The Examiner further states that Narayanaswamy:

> ... does not explicitly teach that the old value and the new value are specified by the user. However, it would have been obvious to one of ordinary skill in the art at the time of invention that the old value and the new value can be specified by the user since the purpose of the invention was to provide the user with a series of input tools or panels for specifying deployment variables and customizing the deployment as needed (see abstract, lines 6-8, col. 5, lines 23-35, and col. 8, lines 37-47).

The Examiner admits that Narayanaswamy and Chan do not teach accepting user specification of a file of the plurality of files and a field, searching the field in the file, and wherein the field is an application configuration information position within the archive file content. The Examiner states that Jackson teaches:

> a resource bundle manager for editing application configuration information (i.e., resource information, see column 1, lines 49-55, column 4, lines 45-59), including a search function which accepts user specification of a file of a plurality of files and a field, searching the field in the file, wherein the field is an application configuration information position within the archive file content (see Fig 12, column 8, lines 28-42).

Jackson does not disclose creating an updated plurality of files by searching for a value and then replacing each occurrence of the value.

G. "compressing the updated plurality of files into a new archive file according to the order and the arrangement stored in the structure file, wherein the order and the arrangement is the same before and after the updated plurality of files are created"

Prior claim 1, to which the Examiner responded, recited "archiving the updated plurality of files into the archive file according to the archive file structure in the structure file; wherein the archive file structure is the order and arrangement of the updated plurality of files within the archive file; and "wherein the application configuration information within the archive file is changed and wherein the archive file structure remains the same before and after the application configuration information is changed." The Examiner cited Narayanaswamy, col. 7, lines 58-61 and col. 8, lines 12-16 and stated "EN: the EAR file is repackaged with the modified deployment descriptors where the JAR file content must be archived according to the original JAR file construct since only the deployment descriptors are modified." The Examiner cites Narayanaswamy, col. 7, lines 58-61 and col. 8, lines 12-16. The Examiner cites Fig. 7, Fig. 8, col. 2, lines 38-41, col. 5, lines 23-35, col. 5, lines 57-col. 16, lines 38-63, and states "EN: the JAR file structure remains the same before and after the data is changed because only the data within the deployment descriptor is changed as a result of the update." The references cited by

the Examiner simply do not disclose the limitations of claim 21 when claim 21 is read a whole because it is the order and arrangement of the files that remains unchanged. The cited references are silent as to the order and arrangement remaining the same. All that is disclosed is a deployment descriptor being modified and then placed in a file. Files are not compressed into a new archive file.

H. The Examiner failed to provide an explicit analysis of his obviousness rejection

In combining references, an explicit analysis is required to combine or modify references. The Supreme Court has stated the following:

> Often, it will be necessary for a court to look to interrelated teachings of multiple patents; the effects of demands known to the design community or present in the marketplace; and the background knowledge possessed by a person having ordinary skill in the art, all in order to determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue. To facilitate review, this analysis should be made explicit. See *In re Kahn*, 441 F. 3d 977, 988 (CA Fed. 2006).[R]ejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness .). As our precedents make clear, however, the analysis need not seek out precise teachings directed to the specific subject matter of the challenged claim, for a court can take account of the inferences and creative steps that a person of ordinary skill in the art would employ. KSR Int'l Co. v. Teleflex Inc., 550 U.S. , 2007 U.S. LEXIS 4745, (2007) at 14. Conclusory statements are insufficient to support obviousness rejections. In particular, "Rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness. See *In re Kahn*, 441 F. 3d 977, 988 (CA Fed. 2006).
> The examiner stated the following in combining the cited references

As can be seen, an express analysis has not been provided in the Examiner's reasons for combining the references. The statements made by the Examiner do not provide reasons as required in the Supreme Court guidance on combining references in the KSR case. Instead, the Examiner has only provided statements that are conclusory or reciting some desired goal. These conclusions and desired goals have not been supported with any explicit analysis or articulated reasoning with some rationale underpinnings to support the conclusions or goals for combining the elements in these two references in the manner proposed by the Examiner. Further, even if these conclusory statements or goals could be considered "reasons", they do not have any articulated reasoning with some rational underpinning to support the examiners assertion of obviousness.

In regard to a user specifying a new value and an old value, the Examiner stated:

> However, it would have been obvious to one of ordinary skill in the art at the time of invention that the old value and the new value can be specified by the user since the purpose of the invention was to provide the user with a series of input tools or panels for specifying deployment variables and customizing the deployment as needed (see abstract, lines 6-8, col. 5, lines 23-35, and col. 8, lines 37-47).

The Examiner has merely offered a goal or desire to "provide the user with a series of input tools or panels for specifying deployment variables and customizing the deployment as needed." To meet this goal, the Examiner states it would have been obvious "that the old value and the new value can be specified by the user since the purpose of the invention was to provide the user with a series of input tools or panels for specifying deployment variables and customizing the deployment as needed." Nowhere, however, has the Examiner gone beyond this conclusion or desired goal to explain sufficiently why one of ordinary skill in the art would have combined these two references.

In regard to the temporary directory, the Examiner stated:

> … it would have been obvious to one or ordinary skill in the art at the time of the invention to have modified Narayanaswamy to create a temporary directory and copy the archive file into the temporary directory as taught by Chan because it provides a temporary working directory for working with the input Jar file." The Examiner cites Chan, col. 7, lines 31-40, col. 7, lines 66 -col. 7, line 15.

The Examiner has merely offered a goal or desire to "provide a temporary working directory for working with the input JAR file." To meet this goal, the Examiner states it would have been obvious to have modified Narayanaswamy to create a temporary directory and copy the archive file into the temporary directory as taught by Chan. Nowhere, however, has the Examiner gone beyond this conclusion or desired goal to explain sufficiently why one of ordinary skill in the art would have combined these two references.

In regard to searching the field in the file, the Examiner stated:

> It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Narayanaswamy and Chan to accept user specification of a file of the plurality of files and a field, searching the field in the file, and "wherein the field is an application configuration information position within the archive file content" as taught by Jackson because it provides context to guide the search and provide an option to the user to search over one particular file (see column 8, lines 35-42 of Jackson.

The Examiner has merely offered a goal or desire to provide "context to guide the search and provide an option to the user to search over one particular file." To meet this goal, the Examiner states it would have been obvious to have modified Narayanaswamy and Chan to accept a user specification of a file of the plurality of files and a field, searching the field in the file, and "wherein the field is an application configuration information position within the archive file content" Nowhere, however, has the Examiner gone beyond this conclusion or desired goal to explain sufficiently why one of ordinary skill in the art would have combined these two references.

Therefore, the rejection of claim 1 under 35 U.S.C. § 103 has been overcome.

III.   Conclusion

It is respectfully urged that the subject application is patentable over the cited reference(s) and is now in condition for allowance.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: December 3, 2008

Respectfully submitted,

/Rudolf O. Siegesmund/

Rudolf O. Siegesmund
Reg. No. 37,720
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicant